AMENDMENTS

IN THE CLAIMS:

Please amend claims 1, 2, 7, 12, 16, 20, 21, 22, and 32, add new claims 33 and 34, and cancel claims 15, 23, and 31.

1. (Currently Amended) A method for transferring data entries from a peripheral to a data queue in a host memory, the method comprising:

determining a lower limit on a number of available data entry positions in the data queue; and

selectively transferring a current data entry to the data queue ~~using a full cache line write~~ if the lower limit is greater than or equal to a first value, wherein the first value is a number related to [[the]] a current cache line size, thereby preventing an overwriting of an unprocessed data entry in the data queue.

2. (Currently Amended) The method of claim 1, wherein the data entries are incoming data status entries, wherein the data queue is an incoming data status ring in the host memory, wherein determining the lower limit on the number of available data entry positions comprises determining a number of available incoming data status entry positions in the incoming data status ring, wherein selectively transferring the current data entry comprises selectively transferring a current incoming data status entry to the host memory ~~using the full cache line write~~ if the lower limit is greater than or equal to the first value~~, and wherein a cache line write is a transfer of data to host memory~~.

3. (Original) The method of claim 2, wherein the lower limit is determined at least in part according to a number of unused incoming data descriptors.

4. (Original) The method of claim 3, wherein the lower limit is determined at least in part according to a number of unused incoming data status entry positions remaining for a current incoming data descriptor.

5. (Original) The method of claim 4, wherein the lower limit is determined at least in part according to a sum of the number of unused incoming data descriptors and the number of unused incoming data status entry positions remaining for the current incoming data descriptor.

6. (Original) The method of claim 5, wherein determining the lower limit comprises calculating a sum of the number of unused incoming data descriptors and the number of unused incoming data status entry positions remaining for the current incoming data descriptor minus 1.

7. (Currently Amended) The method of claim 6, wherein the number of unused incoming data descriptors is [[the]] a difference between an incoming data status pointer and an incoming data descriptor write pointer.

8. (Original) The method of claim 7, wherein the first value is a number of unused incoming data status entry positions remaining in a current cache line.

9. (Original) The method of claim 7, wherein the first value is a number of incoming data status entries per cache line.

10. (Original) The method of claim 3, wherein the first value is a number of unused incoming data status entries positions remaining in a current cache line.

11. (Original) The method of claim 3, wherein the first value is a number of incoming data status entries per cache line.

12. (Currently Amended) The method of claim 3, wherein the number of unused incoming data descriptors is [[the]] a difference between an incoming data status pointer and an incoming data descriptor write pointer.

13. (Original) The method of claim 2, wherein the first value is a number of unused incoming data status entries positions remaining in a current cache line.

14. (Original) The method of claim 2, wherein the first value is a number of incoming data status entries per cache line.

15. (Canceled)

16. (Currently Amended) The method of claim 2, wherein selectively transferring the current incoming data status entry ~~to the host memory using a full cache line write~~ comprises:

determining a first portion of the data queue containing the data status entry and any preceding data status entry transferred to the data queue and a remaining portion of the data queue, wherein the first portion and the remaining portion add up to a full cache line length;

padding the remaining portion of the data.

~~transferring the current incoming data status entry and any previous incoming data status entries for a current cache line to the host memory using a full cache line write~~.

17. (Original) The method of claim 16, wherein the first value is a number of unused incoming data status entries positions remaining in the current cache line.

18. (Original) The method of claim 16, wherein the first value is a number of incoming data status entries per cache line.

19. (Previously Presented) The method of claim 16, wherein determining the lower limit comprises calculating the sum of a number of unused incoming data descriptors and a number of unused incoming data status entry positions remaining for a current incoming data descriptor minus 1.

20. (Currently Amended) A system for transferring incoming data status entries from a peripheral to a host memory, comprising:

a descriptor management system in the peripheral, the descriptor management system adapted to determine a lower limit, a number of available incoming data status entry positions in an incoming data status ring in the host memory, and to selectively transfer a current incoming data status entry to the data status ring ~~using a full cache line write~~ if the lower limit is greater than or equal to a first value, wherein the first value is a number related to the current cache line size, thereby preventing an overwriting of an unprocessed data entry in the host memory ~~data queue~~.

21. (Currently Amended) A peripheral system for providing an interface between a host computer and an external device or network, the peripheral system comprising:

a descriptor management system adapted to determine a lower limit, a number of available data entry positions in a data queue in a host memory, and to selectively transfer a current data entry to the data queue ~~using a full cache line write~~ if the lower limit is greater than or equal to a first value, wherein the first value is a number related to [[the]] a current cache line size, thereby preventing an overwriting of an unprocessed data entry in the data queue.

22. (Currently Amended) The system of claim 21, wherein ~~the~~ data entries are incoming data status entries, wherein the data queue is an incoming data status ring in the host memory, and wherein the descriptor management system is adapted to determine [[a]] the lower limit on a number of available incoming data status entry positions in the incoming data status ring, and to selectively transfer a current incoming data status entry to the host memory ~~using the full cache line write~~ if the lower limit is greater than or equal to the first value.

23. (Canceled)

24. (Original) The system of claim 22, wherein the descriptor management system determines the lower limit at least in part according to a number of unused incoming data descriptors.

25. (Original) The system of claim 24, wherein the descriptor management system determines the lower limit at least in part according to a number of unused incoming data status entry positions remaining for a current incoming data descriptor.

26. (Original) The system of claim 25, wherein the descriptor management system determines the lower limit at least in part according to a sum of the number of unused incoming data descriptors and the number of unused incoming data status entry positions remaining for the current incoming data descriptor.

27. (Original) The system of claim 26, wherein the descriptor management system determines the lower limit as a sum of the number of unused incoming data descriptors and the number of unused incoming data status entry positions remaining for the current incoming data descriptor minus 1.

28. (Previously Presented) The system of claim 27, wherein the descriptor management system determines the number of unused incoming data descriptors as a difference between an incoming data status pointer and an incoming data descriptor write pointer.

29. (Original) The system of claim 22, wherein the first value is a number of unused incoming data status entry positions remaining in a current cache line.

30. (Original) The system of claim 22, wherein the first value is a number of incoming data status entries per cache line.

31. (Canceled)

32. (Currently Amended) The system of claim 22, wherein the peripheral system is a network controller adapted to provide an interface between a host computer and a network.

33. (New) The method of claim 1, wherein selectively transferring a current data entry to the data queue is performed by writing the current data to equal a full cache line length.

34. (New) The method of claim 21, wherein the descriptor management system is further configured to selectively transfer the current data entry to the data queue by writing the current data entry to equal a full cache line length.